PCT

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification [7] : **H04L 29/06, H04N 7/24**	**A1**	(11) International Publication Number: **WO 00/11847** (43) International Publication Date: 2 March 2000 (02.03.00)

(21) International Application Number: PCT/EP99/05775

(22) International Filing Date: 6 August 1999 (06.08.99)

(30) Priority Data:
09/138,782   24 August 1998 (24.08.98)   US

(71) Applicant: KONINKLIJKE PHILIPS ELECTRONICS N.V. [NL/NL]; Groenewoudseweg 1, NL–5621 BA Eindhoven (NL).

(72) Inventors: MALLART, Raoul; Prof. Holstlaan 6, NL–5656 AA Eindhoven (NL). SINHA, Atul; Prof. Holstlaan 6, NL–5656 AA Eindhoven (NL).

(74) Agent: FAESSEN, Louis, M., H.; Internationaal Octrooibureau B.V., Prof. Holstlaan 6, NL–5656 AA Eindhoven (NL).

(81) Designated States: BR, CA, CN, IN, JP, KR, MX, TR, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: EMULATION OF STREAMING OVER THE INTERNET IN A BROADCAST APPLICATION
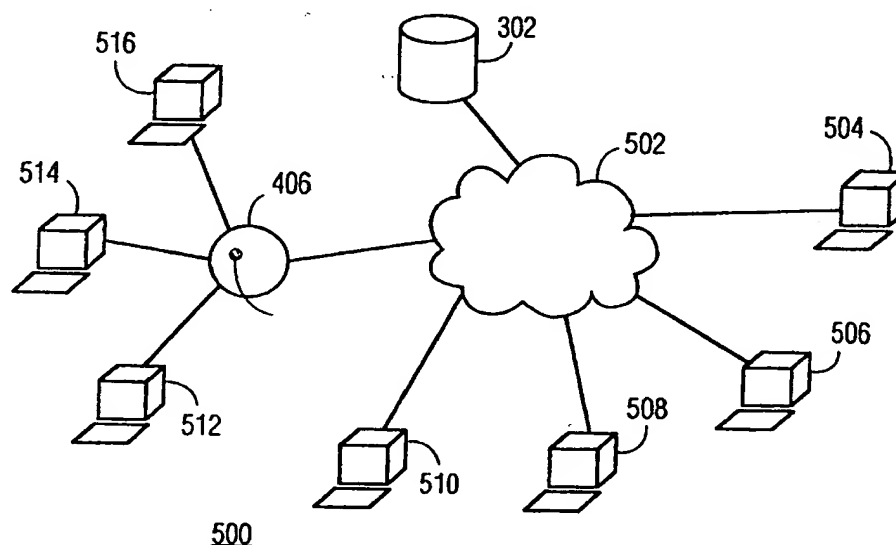
(57) Abstract

In a broadcast application on a client–server network the streaming is emulated of animation data over the Internet to a large number of clients. The animation is considered a sequence of states. State information is sent to the clients instead of the graphics data itself. The clients generate the animation data itself under control of the state information. The server and clients communicate using a shared object protocol. Thus, streaming is accomplished as well as a broadcast without running into severe network bandwidth problems.

Emultation of streaming over the internet in a broadcast application.

FIELD OF THE INVENTION

The invention relates to streaming multimedia files via a network. The invention relates in particular to enabling the emulation of streaming graphics or video animation over the Internet within a broadcast context.

5

BACKGROUND ART

The term "streaming" refers to transferring data from a server to a client so that it can be processed as a steady and continuous stream at the receiving end. Streaming technologies are becoming increasingly important with the growth of the Internet because

10   most users do not have fast enough access to download large multimedia files comprising, e.g., graphics animation, audio, video, or a combination thereof, etc. Streaming, however, enables the client's browser or plug-in to start processing the data before the entire file has been received. For streaming to work, the client side receiving the file must be able to collect the data and send it as a steady stream to the application that is processing the data. This

15   means that if the client receives the data faster than required, the excess data needs to be buffered. If the data does not arrive in time, on the other hand, the presentation of the data will not be smooth.

The term "file" is used herein to indicate an entity of related data items available to a data processing and capable of being processed as an entity. Within the context

20   of the invention, the term "file" may refer to data generated in real-time as well as data retrieved from storage.

Among the technologies that are currently available or under development for the communication of graphics data via the Internet are VRML 97 and MPEG-4. VRML 97 stands for "Virtual Reality Modeling Language", and is an International Standard (ISO/IEC

25   14772) file format for describing interactive 3D multimedia content on the Internet. MPEG-4 is an ISO/IEC standard being developed by MPEG (Moving Picture Experts Group). In both standards, the graphical content is structured in a so-called scene graph. A scene graph is a family tree of coordinate systems and shapes, that collectively describe a graphics world. The top-most item in the scene family tree is the world coordinate system. The world coordinate

system acts as the parent for one or more child coordinate systems and shapes. Those child coordinate systems are, in turn, parents to further child coordinate systems and shapes, and so on.

VRML is a file format for describing objects. VRML defines a set of objects

5 useful for doing 3D graphics, multi-media, and interactive object/world building. These objects are called nodes, and contain elemental data which is stored in fields and events. Typically, the scene graph comprises structural nodes, leaf nodes, interpolation nodes and sensor nodes. The structural nodes define the spatial relationship of objects within a scene. The leaf nodes define the physical appearance of the objects. The interpolation nodes define

10 animations. The sensor nodes define user interaction for particular user input modalities. VRML does not directly support streaming of data from a server into a client. Facilities such as synchronization between streams and time stamping that are essential in streaming do not exist in VRML. However, VRML has a mechanism that allows external programs to interact with VRML clients. This has been used in sports applications to load animation data into the

15 client. See, for example, "VirtuaLive Soccer" of Orad Hi-Tec Systems, Ltd at <http://www.virtualive.com>. This web document discusses a process for producing realistic, animated, three-dimensional graphic clips that simulate actual soccer match highlights for being sent via the Internet. The system generates content that complements television sports coverage with multimedia-rich Web pages in near real time. In this example, the process

20 works in two steps. First the graphics models of the stadium and of the soccer players are downloaded along with an external program, in this case a Java Applet. The user can then interact with the external program to request a particular animation. The data for this animation is then downloaded into the client and interacted with by the user. In terms of node type, this process first downloads the structural and leaf nodes, and thereupon the interpolation

25 nodes. By changing the set of interpolation nodes, it is possible to run a different animation sequence. The process used in this example is somewhat equivalent to a single step process in which the user can choose the complete VRML file that contains all the models (structural nodes) and all the animation data (interpolator nodes). This approach leads to long download times before any content can be played on the client. This is experienced as a frustrating

30 experience, especially if compared to TV broadcast where content is available instantly.

The other technology introduced above, MPEG-4, defines a binary description format for scenes (BIFS) that has a wide overlap with VRML 97. MPEG-4, on the other hand, has been designed to support streaming of graphics as well as for video. MPEG-4 defines two server/client protocols for updating and animating scenes: BIFS-Update and BIFS-Anim.

Some of the advantages of MPEG-4 over VRML are the coding of the scene description and of the animation data as well as the built-in streaming capability. The user does not have to wait for the complete download of the animation data. For example, in the soccer match broadcast application mentioned earlier the animation can start as soon as the models of the

5 players and the stadium are downloaded. MPEG-4 further has the advantage that it is more efficient owing to its BIFS transport protocol that uses a compressed binary format.

Within the context of streaming, the known technologies mentioned above have several limitations with regard to bandwidth usage, packet-loss concealment or recovery and multi-user interactivity, especially in a broadcast to large numbers of clients.

10 As to bandwidth, the complete animation is generated at the server. This results in a large amount of data that needs to be transported over the network, e.g., the Internet, connecting the client to the server. For example, in the soccer broadcast application mentioned above, the 22 soccer players need to be animated. Each animation data point per individual player comprises a position in 3D space and a set of, say, 15 joint rotations to model the

15 player's posture. This represents 63 floating-point values. If it is assumed that the animation update rate is 15 data points per seconds, a bit-rate of 665 Kbps is required. This bit-rate can be reduced through compression. Typically, using BIFS reduces the bit-rate by a factor of 20, giving a bit-rate of about 33 Kbps. However, this number has not taken into account overhead required for the Internet protocols (RTP, UDP and IP) and for additional data types, such as

20 audio. However, typical modems currently commercially available on the consumer market have a capacity of 28.8 Kbps or 33.6 Kpbs. It is clear that streaming animation causes a problem at the end user due to bandwidth limitations. In the case of a broadcast to a large number of clients, say 100,000 clients, the data stream will need to be duplicated at several routers. A router on the Internet determines the next network point to which a packet should be

25 forwarded on its way toward its final destination. The router decides which way to send each information packet based on its current understanding of the state of the networks it is connected to. A router is located at any juncture of networks or gateway, including each Internet point-of-presence. It is clear that the broadcast could lead to an unmanageable data explosion across the Internet. To prevent that from happening, the actual bandwidth needs to

30 be limited to much lower than 28.8 Kbps.

As to packet loss concealment, VRML-based systems utilize reliable protocols (TCP). Packet losses are not an issue here. In the case of MPEG-4, BIFS uses RTP/UDP/IP. A packet-loss recovery mechanism is therefore required. In a point-to-point application, re-transmission of lost packets can be considered. In a broadcast situation, however, this is much

more complex. In both cases, however, MPEG reliability requires either higher bandwidth

usage (redundancy) or higher latency (retransmission).

As to multi-user interactivity, both VRML and MPEG-4 are essentially based

on a server-client communication. No provisions exist to enable communication among

5   multiple clients.

For more information on VRML see, for example, "Key Concepts", March 5,

1996, at: <http://sgi.felk.cvut.cz/~holecek/VRML/concepts.html>, or "Internetwork

Infrastructure Requirements for Virtual Environments", D.P. Brutzman et al., Jan.23, 1996,

publicly available at: <http://www.stl.nps.navy.mil/~brutzman/vrml/vrml_95.html>.

10   For more information on MPEG-4 see, for example, "Overview of the MPEG-4

Standard", ISO/IEC JTC1/SC29/WG11 N2323 ed. Rob Koenen, July 1998, publicly

available at <http://drogo.cselt.stet.it/mpeg/standards/mpeg-4/mpeg-4.htm>.

OBJECT OF THE INVENTION

15   It is therefore an object of the invention to provide a technology that enables a

client to process multimedia data as if it were a steady and continuous stream. It is another

object to enable the continuous processing at a large number of clients in a broadcast over the

Internet. It is noted that the problems identified above become rather acute in a broadcast

application.

20

SUMMARY OF THE INVENTION

To this end, the invention provides a method of emulating streaming a

multimedia file via a network to a receiving station connected to the network. Respective state

information descriptive of respective states of the file is supplied. The receiving station is

25   enabled to receive the respective state information via the network and is enabled to locally

generate the multimedia file under control of the respective state information. In a broadcast

for animation, the invention relates to a method of supplying data via a network for enabling

displaying graphics animation. Respective state information is supplied over the network

descriptive of successive respective states of the animation. The respective state information is

30   received via the network. The receiving station is enabled to generate the animation under

control of the respective state information upon receipt.

In the invention the multimedia file (animation, video or audio file) is described

as a succession of states. It is this state information that gets transmitted to the clients rather

than the animation data itself. The term "emulating" therefore emphasizes that the information

communicated to the client need not be streamed. The client generates the data for play-out
locally and based on the state information received. Accordingly, the user perceives a
continuous and steady stream of data during play-out as if the data were streamed over the
network (under optimal conditions).

5   In a preferred embodiment, a shared-object protocol is used to accomplish the
emulation. Both a server and a client have copies of a collection of objects. An object is a data
structure that holds state information. Within the context of the virtual soccer match, an object
is, for example, a graphics representation of one of the soccer players. The server receives a
streamed video file and monitors the objects. It is noted that MPEG-4 enables the creation of
10   video objects that are processed as an entity. If the server changes the state of this object, the
shared object protocol causes the copy at the client to change accordingly. This is explained in
more detail with reference to the drawings.

           This state information is at a higher level of abstraction than the animation data
itself. For example, in the soccer match broadcast application mentioned above, the state
15   information comprises the current positions of the 22 players in the field and parameters
specifying their current action (e.g., "running", "jumping", etc.). The use of higher level
information has several advantages, in particular in a broadcast application where animation
is streamed over the Internet to a large audience. The content of the state information as
communicated over the Internet is very compact, thus requiring lower bandwidth than in case
20   the animation data itself is streamed. The animation is generated locally from a few
parameters. In addition, the update rate of animation data points is lower because the state of
the animation changes at a slower rate than the animation data itself. This contributes to
further lowering bandwidth requirements. Furthermore, the invention provides enhanced
possibilities for packet loss recovery or concealment and for network latency jitter masking. It
25   is easy to interpolate or extrapolate between states and to implement dead reckoning concepts.
User interaction with the animation is more easily programmable because of this higher level
of abstraction. Another advantage is that multi-user interaction is feasible if clients are enabled
to share state information. Still another advantage is the fact that clients are enabled to convert
the state information into animation based on their individual processing power that might
30   differ from client to client. The resources available at the client may be different per client or
groups of clients.

           Within the context of the invention, reference is made to U.S. Patent
Application Serial No. 09/053,448 (PHA 23,383) of same Assignee, titled "Group-wise video
conferencing uses 3D-graphics model of broadcast event" and incorporated herein by

reference. This document addresses a TV broadcast service to multiple geographically distributed end users. The broadcast service is integrated with a conferencing mode. Upon a certain event in the broadcast, specific groups of end users are switched to a conference mode under software control so that the group is enabled to discuss the event. The conference mode

5   is enhanced by a 3D graphics model of the video representation of the event that is downloaded to the groups. The end-users are capable of interacting with the model to discuss alternatives to the event.

BRIEF DESCRIPTION OF THE DRAWINGS

10   The invention is explained by way of example and with reference to the accompanying drawings, wherein:

Fig.1 is a diagram of a VRML client-server system;

Fig.2 is a diagram of an MPEG-4 client-server system; and

Figs.3-6 are diagrams of systems in the invention.

15   Throughout the figures, same reference numerals indicate similar or corresponding features.

PREFERRED EMBODIMENTS

Fig.1 is a block diagram of a client-server system 100 based on VRML. System

20   100 comprises a server 102 coupled with a client 104 via a communication channel 106, here the Internet. System 100 may comprise more clients but these are not shown in order to not obscure the drawing. Server 102 comprises a source encoder 108 and a channel encoder 110. Client 104 comprises a channel decoder 112 and a source decoder 114. Source encoder 108 is considered a content generation tool. For example, it can be a tool that generates the VRML

25   animation data from motion capture devices (not shown) operating on video. Channel encoder 110 is a sub-system that takes as input the VRML animation generated at source encoder 108 and transforms it into a form that can be transported over the Internet. The VRML animation data is stored in a file. The transport of this file uses a standard file transport protocol. At client 104, channel decoder is contained in an external program 116. It gets the animation data

30   from the downloaded file and sends it to a VRML player 118 that performs the source decoder function. The source decoder function is essentially a management of the scene graph. This server-client communication procedure is not a streaming solution. The specification of VRML does not consider streaming a requirement. Facilities such as synchronization between streams and time stamping, both essential to streaming, do not exist in VRML.

Fig. 2 is a block-diagram of a client-server system 200 based on MPEG-4. System 200 has a server 202 coupled to a client 204 via a communication channel 206. Server 202 has a source encoder 208 and a channel encoder 210. Client 204 has a channel decoder 212 and a source decoder 214. As mentioned above, MPEG-4 has been designed to support
5   streaming. Among other things, MPEG-4 has defined a binary description format for scenes (BIFS) that has a wide overlap with VRML 97. Furthermore, MPEG-4 defines two server/client protocols for updating and animating scenes, namely BIFS-Update and BIFS-Anim. Advantages of MPEG-4 over VRML within the context of streaming is the coding of the scene description and of the animation data as well as the built-in streaming
10 capability. Source encoder 208 is, similarly to encoder 108, a content generation tool. Channel encoder 210 is different from channel encoder 110. It generates a bit-stream in BIFS and BIFS-Anim formats. This bit-stream contains the graphics models of the players and the stadium (in the soccer match animation) as well as the animation data.

However, both systems 100 and 200 have several serious drawbacks when used
15 in an environment for broadcasting animation to a large number of clients, say, 100 - 100,000 clients. The limitations relate to network bandwidth usage, packet loss concealment and multi-user interactivity as already mentioned above.

A preferred embodiment of the invention provides a solution to these problems by emulating the streaming utilizing a communication protocol that supports the sharing of
20 objects by an object owner and an object viewer (or listener). A shared object is a data structure that holds state information. The set of shared objects that defines the entire state is called the world model. The clients and the server have their own copy of the world model. For example, an object within the context of a soccer match representation is the representation of a soccer player. The object's state information is then, for example, the
25 position of the soccer player in 3D space or an action state such as "running" or "jumping" or "sliding" or "lying on the ground, apparently hurt, but has track record of comedian". Each shared object is owned by a particular party, e.g., the server. The owner can change the state information contained in the object. When this occurs, the protocol automatically synchronizes the state information across the network. Such a protocol is referred to herein below as a
30 shared-object support protocol. The protocol ensures that all the world model copies remain consistent as the state of the world model evolves. Examples of protocols that can be used for this purpose are DIS (Distributed Interactive Simulation) and ISTP (Interactive Sharing Transfer Protocol). An underlying idea of the invention is to describe the animation as a succession of states. For example, in the soccer software application, the animation is

described as a succession of player positions on the field and action states of the players. The state at a given moment is represented by the world model. As time passes, the state evolves and the protocol synchronizes the state of the world model across the network. This can also be explained in terms of shared objects. These objects hold the state information that describes

5   the game at a given time instant. Updates of the state information for each object result in the generation of messages that are being sent across the network to the clients.

Fig.3 is a block diagram of a system 300 in the invention. System 300 comprises a server 302 coupled to a client 304 via a network 306. Server 302 comprises a source encoder 308 and a channel encoder 310. Client 304 comprises a channel decoder 312

10   and a source decoder 314. Server 302 has a copy 316 of a world model and client 304 has a copy 318 of the world model. Data is streamed to source encoder 308 at an input 320. Source encoder 308 generates the required state information based on the input received and updates the state of the objects in world model copy 316 as the streaming process continues. This type of technology is used, for example, by the VirtuaLive Soccer system mentioned above.

15   Channel encoder 310 monitors world model copy 316 and encodes the state changes of the shared objects. The encoded state changes are sent to client 304 via network 306. Channel decoder receives the state changes and updates local world model copy 318. Source decoder 314 performs two tasks. First, it generates the animation based on the state information received. Secondly, source decoder 314 manages the scene graph according to the animation.

20   Source decoder 314 is now an intelligent component: it performs animation computation and, in addition, it is capable of performing other tasks such as state interpolation or extrapolation to conceal packet losses or network latency jitter.

Within this context reference is made to U.S. Patent Application Serial No. 08/722,414 (PHA 23,155) of same Assignee, titled "Multi-player video game with local

25   updates mitigates latency effects", herein incorporated by reference. This reference discusses a system wherein multiple users share a virtual environment through an interactive software application. State changes of a specific user are transmitted to one or more other users, dependent on respective relative distances in the virtual environment between the specific user and each respective one of the other users. This conditional transmission reduces message

30   traffic and allows the virtual environment to scale practically indefinitely.

Also, reference is made to U.S. Patent Application Serial No. 08/722,413 (PHA 23,156) of same Assignee, titled "Latency effect in multi-player video game reduced by surrogate agent" and incorporated herein by reference. This document relates to a data processing system that processes an interactive software application for a competition between

two or more users. The system comprises user-interface machines for actuation by respective ones of the users. The machines are interconnected via a network. In order to effectively eliminate latency, a user is represented at the other users' machines as an agent whose reaction to an action of the other user is governed by a rule base stored in the system.

Further reference is made to U.S. Patent Application Ser. No. 08/994,827 (PHA 23,319) of same Assignee and incorporated herein by reference, titled "Diversion agent uses cinematographic techniques to mask latency". This document relates to a software agent as a functional part of a user-interactive software application running on a data processing system. The agent creates a user-perceptible effect in order to mask latency present in delivery of data
10   to the user. The agent creates the effect employing cinematographic techniques.

It is noted that copies 316 and 318 of the world model need not be identical, e.g., in appearance when rendered, as long as an object in one copy of the world model and another object in another copy of the world model are being treated as shared in the sense that they share state changes. The feasibility and extent of non-identity is application-dependent.
15   For example, if one client's user likes to represent the soccer players as, say, penguins, and another client's user prefers the representation of say, ballet dancers, the representation at both clients are being kept consistent throughout the system by means of the shared state changes. As another example, client 304 may enable the user to input additional state information to control the rendering of the world model at play-out. For example, the user may select a
20   particular point of view when watching the VirtuaLive soccer match. This state information is not, and need not, be present in server 302. It is noted that the rendering of the viewpoint based on state information and the world model is much less complicated and requires fewer resources than if the image were actually streamed into client 304 as a bitmap with depth information. Accordingly, in addition to the advantages of the invention mentioned earlier, the
25   invention facilitates user-interactivity.

The configuration of system 300 assumes that client 304 is capable of running a software application and has a sufficiently powerful CPU and a sufficiently large storage. Some clients may not have these capabilities onboard. It is therefore desirable to consider lower-end terminals, also referred to as "thin clients". Such terminals could be, for example,
30   low-profile MPEG-4 terminals that accept a BIFS stream as input but may not be powerful enough. This is explained with reference to Fig.4 that is a block diagram of a system 400 according to the invention. System 400 comprises a server 302 that communicates with client 204 via a translator station 406. The configuration of server 302 and client 204 has been discussed above. Translator station 406 maintains a local copy of the world model. This world

model is updated by the messages from server 302 so that the model represents the current state. Based on this state information, translator station 406 computes the animation. The animation data is encoded in BIFS-Anim format and transmitted to MPEG-4 client 204. Server 302 is similar to the one in system 300. Translator station 406 is a module that performs a

5 conversion between messages transmitted under the shared-object support protocol on the one hand, and the BIFS-Anim bit-stream on the other hand. Station 406 has a channel decoder 312 discussed above, a source transcoder 410 and a channel encoder 412. Decoder 312 interprets the messages received from server 302 and updates local copy of the world model 318. Source transcoder 410 comprises a program that computes the animation based on state information.

10 This module preferably performs such tasks as packet loss recovery (based on interpolation or extrapolation), dead reckoning, local animation, etc, similar to source decoder 314 above. Channel encoder 412 generates a bit-stream in BIFS and BIFS-Anim formats based on the output of source transcoder 410.

Fig.5 is a block diagram of a system 500 in the invention. System 500 combines

15 the configurations of systems 300 and 400. System 500 comprises a server 302, a network 502, clients 504, 506, 508 and 510 connected to server 302 via network 502. System 500 further comprises a translator station 406 and clients 512, 514 and 516. Clients 512-516 are coupled to server 302 via network 502 and translator station 406. Clients 512-516 are served by translator station 406 with BIFS bit streams, whereas clients 504-510 receive the state

20 information in a protocol that supports shared objects and generate the animation themselves.

Fig.6 is a block diagram of a system 600 in the invention that enables interaction among clients. System 600 comprises a server 302 coupled to clients 602 and 604 via network 606. The configuration of server 302 is discussed above. Server 302 has a copy of a world model with objects 608, 610, 612 and 614. Clients 602 and 604 have similar copies of

25 the world model with similar objects 608-614. The copies of the world model are maintained consistent throughout system 600 through state information sent by server 302. This forms the basis for the emulation of streaming a graphics animation, a video animation or an audio file as discussed above. Clients 602 and 604 now also share objects 616 and 618 with each other, but not with server 302. For example, client 602 is owner of an object "viewpoint" that

30 represents the view of the graphics representation of the soccer match chosen by client 602. Based on the state information received from server 302 client 602 renders a graphics image of the match as if it were viewed from a particular position in the stadium. The rendering of the image is based on the combination of current state information received from server 302, the local world model copy and the user input via user input means 620, e.g., a joystick or mouse,

that enables selection of the viewpoint. Client 604 shares the viewpoint object that is being kept consistent with the one at client 602 under the control of the latter and using the shared-object support protocol. Objects 616-618 are not shared with other clients on the system. It is noted that the rendering of the viewpoint based on state information and the world model is

5 much less complicated and requires fewer resources than if the image were actually streamed into clients 602 and 604 as a bitmap with depth information. System 600 may even be a totally distributed system without a server with main ownership. Each respective one of multiple clients owns respective ones of the objects in the world model that is perceptible from all clients. The owner of an object triggers a state change that gets propagated through the

10 network so as to maintain consistency within the shared world model. In a multi-user application the effect is a continuous play-out at each client without severe bandwidth limitations as a consequence of the emulation of streaming of the animation.

CLAIMS:

1.   A method of emulating streaming a multimedia file via a network to a receiving station connected to the network, the method comprising:
- supplying respective state information descriptive of respective states of the file;
- enabling the station to receive the respective state information via the network; and
5   - enabling the station to generate the file under control of the respective state information.

2.   The method of claim 1, comprising:
- receiving the file through streaming; and
- generating the respective state information based on the file received.

10

3.   The method of claim 2, comprising using a shared-object protocol to communicate the state information to the station.

4.   The method of claim 1, wherein the respective state information is broadcasted
15   over the network to multiple receiving stations.

5.   The method of claim 1, wherein the file comprises graphics animation.

6.   The method of claim 1, wherein the file comprises audio.

20

7.   The method of claim 1, wherein the file comprises video.

8.   A method of supplying data via a network for enabling playing out a file, the method comprising:
25   - supplying over the network respective state information descriptive of successive respective states of the file;
- enabling receiving the respective state information via the network; and
- enabling generating the file under control of the respective state information upon receipt.

9.   The method of claim 8, comprising supplying the data using a shared object protocol.

10.   The method of claim 9, comprising supplying the data in a broadcast.

11.   A station for use in a server-client system, wherein:
- the server is coupled to at least one client;
- the system is capable of emulating streaming a multimedia file via a network to the station;
- the server supplies respective state information descriptive of respective states of the file;
- the station is capable of receiving the respective state information via the network;
- the station is capable of generating the file under control of the respective state information; and
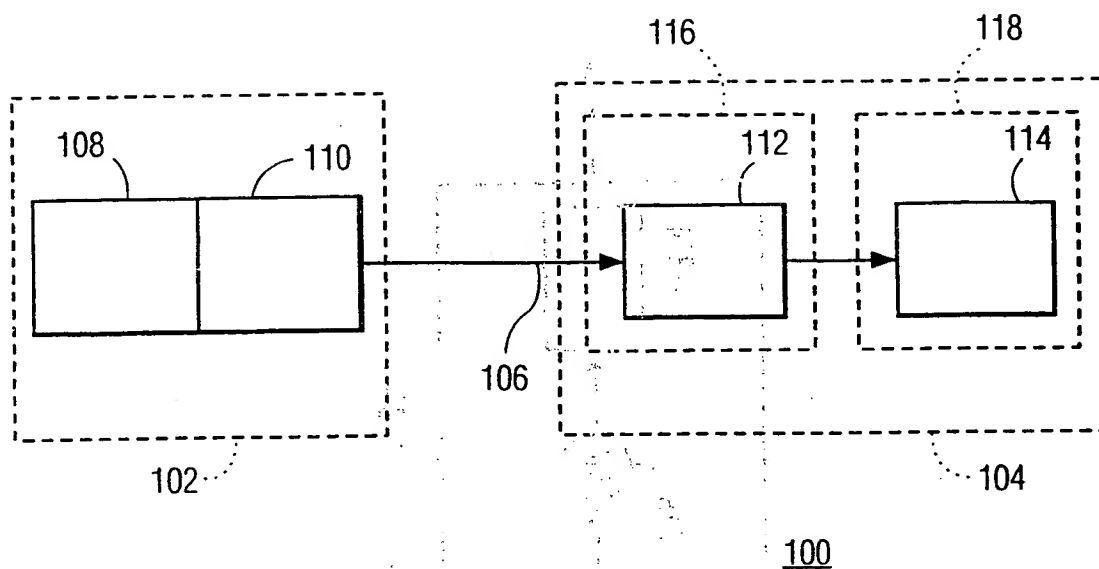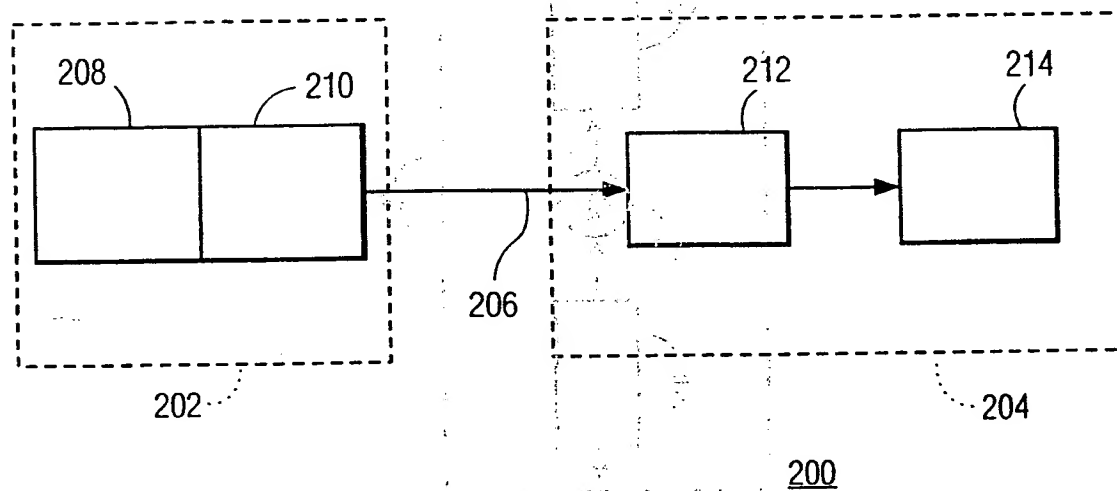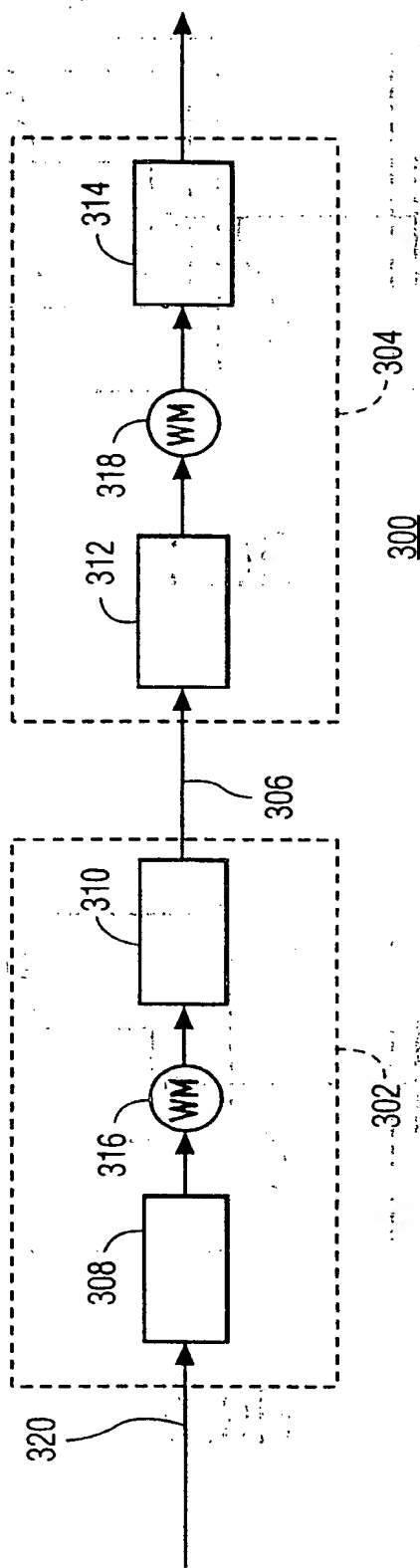- the station streams the file generated to the at least one client.
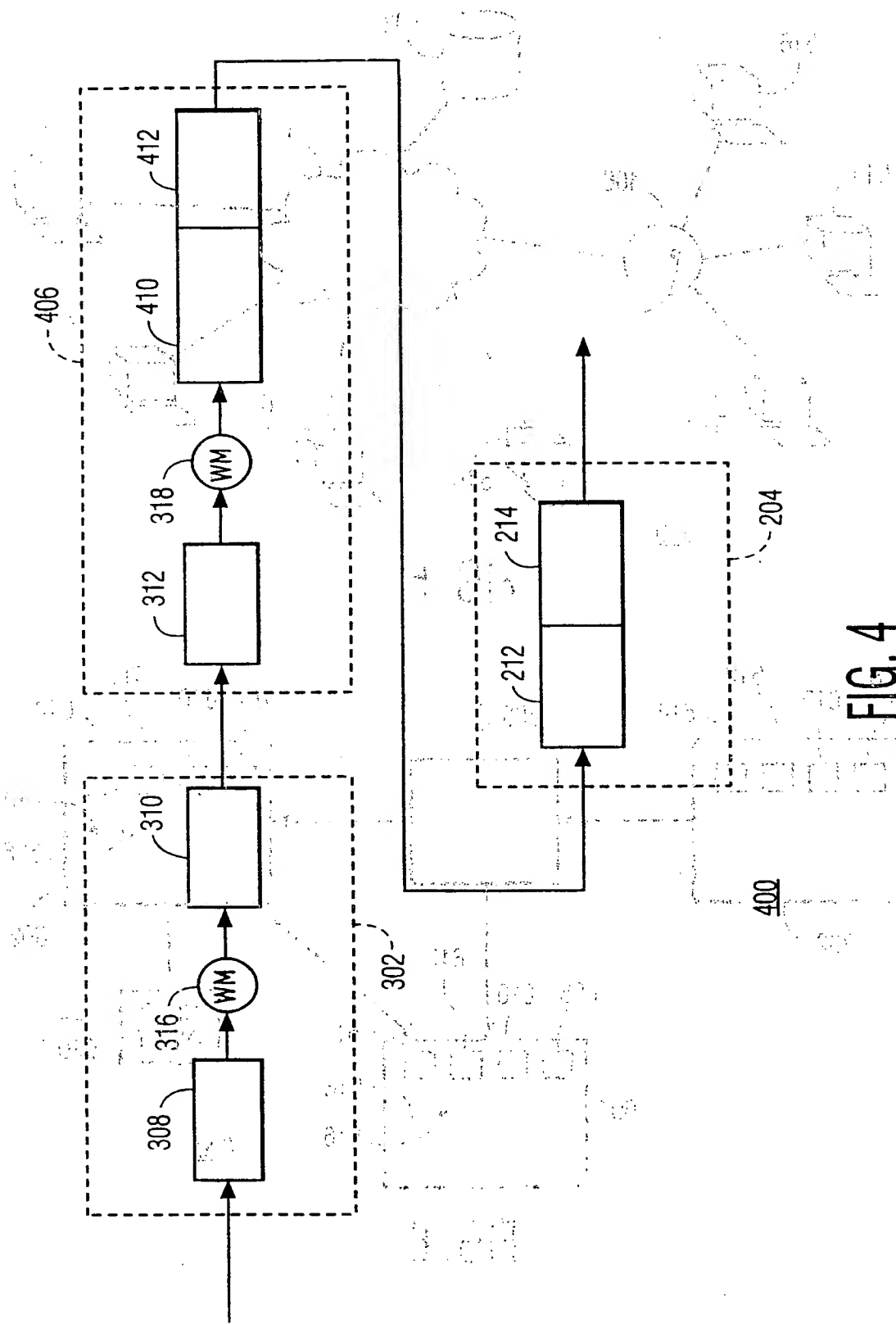
FIG. 1

100

FIG. 2

200

2/4

FIG. 3

FIG. 4

516

302

514

406

502

504

506

508

510

512

500

FIG. 5

608 610 612 614

606

608 610 612 614

618
616

302

612

602

608 610 614

618
616

USER
INPUT

620

604

600
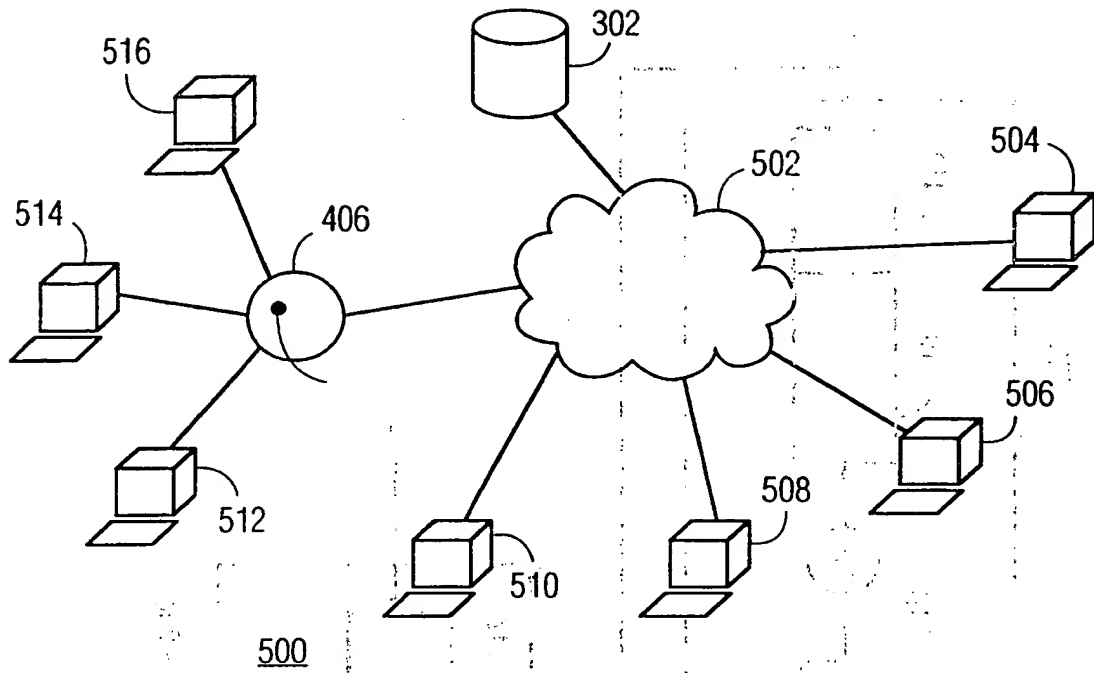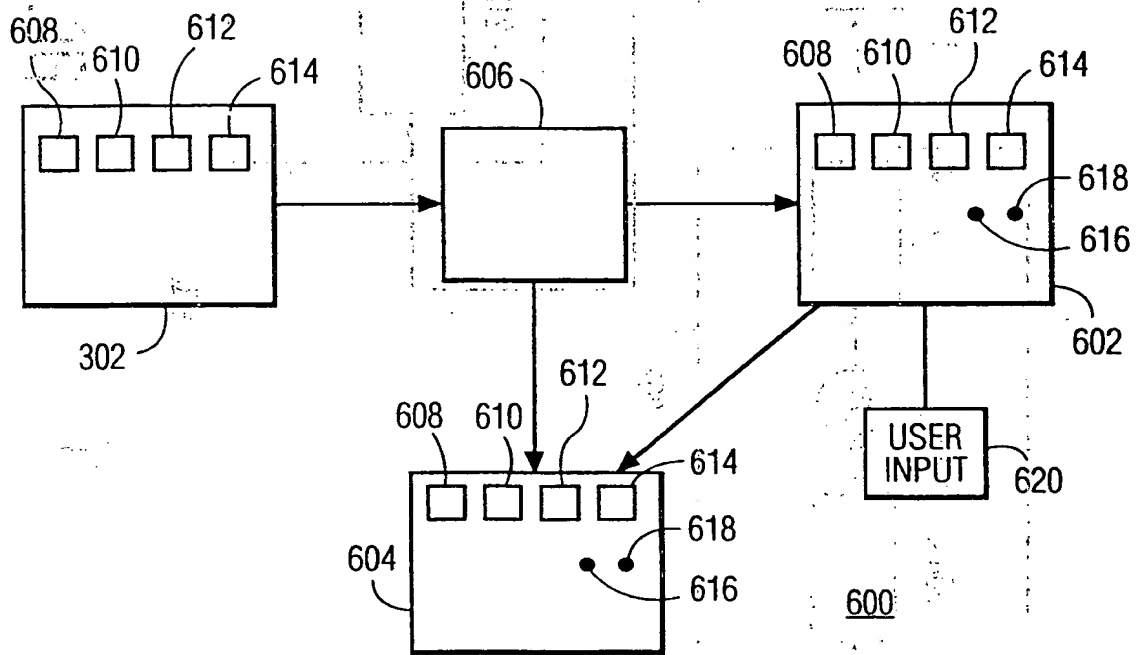
FIG. 6

A. CLASSIFICATION OF SUBJECT MATTER

IPC 7   H04L29/06   H04N7/24

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7   H04N   G06F   H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category°	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P,X	EP 0 909 069 A (MITSUBISHI ELECTRIC CORP) 14 April 1999 (1999-04-14) the whole document	1-11
X	JP 09 307590 A (MITSUBISHI ELECTRIC RES LAB INC) 28 November 1997 (1997-11-28)	1-11
P,X	& US 5 842 216 A (ANDERSON DAVID ET AL.) 24 November 1998 (1998-11-24) the whole document	

[X] Further documents are listed in the continuation of box C.   [X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
17 December 1999	13/01/2000

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Giannotti, P

6

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
X	WATERS, R.; ANDERSON, D.B.; SCHWENKE, D.L.: "Design of the Interactive Sharing Transfer Protocol" PROCEEDINGS OF THE SIXTH IEEE WORKSHOPS ON ENABLING TECHNOLOGIES: INFRASTRUCTURE FOR COLLABORATIVE ENTERPRISES, 1997. 18 - 20 June 1997, pages 140-147, XP002125528 the whole document	1-11
A	EP 0 753 834 A (SONY CORP) 15 January 1997 (1997-01-15) the whole document	1-11
A	KESSLER G D ET AL: "A NETWORK COMMUNICATION PROTOCOL FOR DISTRIBUTED VIRTUAL ENVIRONMENT SYSTEMS" PROCEEDINGS OF THE IEEE VIRTUAL REALITY ANNUAL INTERNATIONAL SYMPOSIUM, US, LOS ALAMITOS, IEEE COMP. SOC. PRESS, 30 March 1996 (1996-03-30) - 3 April 1996 (1996-04-03), page 214-221 XP000669965 ISBN: 0-8186-7296-X the whole document	1-11
A	DOENGES P K ET AL: "AUDIO/VIDEO AND SYNTHETIC GRAPHICS/AUDIO FOR MIXED MEDIA" SIGNAL PROCESSING. IMAGE COMMUNICATION,NL,ELSEVIER SCIENCE PUBLISHERS, AMSTERDAM, vol. 9, no. 4, - May 1997 (1997-05) page 433-463 XP000700948 ISSN: 0923-5965	
A	ARIKAWA M ET AL: "DYNAMIC LOD FOR QOS MANAGEMENT IN THE NEXT GENERATION VRML" PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON MULTIMEDIA COMPUTING AND SYSTEMS,US,LOS ALAMITOS, CA, 17 - 23 June 1996, page 24-27 XP002060163	
A	KATKERE A ET AL: "Interactive video on WWW: Beyond VCR-like interfaces" COMPUTER NETWORKS AND ISDN SYSTEMS,NL,NORTH HOLLAND PUBLISHING. AMSTERDAM, vol. 28, no. 11, May 1996 (1996-05), page 1559-1572 XP004018251 ISSN: 0169-7552	

-/--

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	ED. ROB KOENEN : "Overview of the MPEG-4 Standard " ISO/IEC JTC1/SC29/WG11 MPEG97/N2323 , - July 1998 (1998-07) pages 1-55 XP002120014 cited in the application	

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
EP 0909059	A	14-04-1999	JP	I1161622 A	18-06-1999
JP 9307590	A	28-11-1997	US	5842216 A	24-11-1998
EP 0753834	A	15-01-1997	CA	2180904 A	12-01-1997
			JP	9081495 A	28-03-1997

THIS PAGE BLANK (USPTO)